Title

Toward Fast and Reliable Potential Energy Surfaces for Metallic Pt Clusters by Hierarchical Delta Neural Networks.

Journal

Journal of chemical theory and computation, 15(10)

ISSN

1549-9618

Authors

Sun, Geng
Sautet, Philippe

Publication Date

2019-10-01

DOI

10.1021/acs.jctc.9b00465

Towards fast and reliable potential energy surfaces for metallic Pt clusters by hierarchical delta neural networks

Geng Sun[1], Philippe Sautet*[1,2]

1. Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, Los Angeles, California 90095, United States

2. Department of Chemistry and Biochemistry, University of California, Los Angeles, Los Angeles, California 90095, United States

sautet@ucla.edu

Abstract

Data-driven machine learning force fields (MLF) are more and more popular in atomistic simulations, and exploit machine learning methods to predict energies and forces for unknown structures based on the knowledge learned from an existing reference database. The latter usually comes from density functional theory calculations. One main drawback of MLFs is that physical laws are not incorporated in the machine learning models and instead, MLFs are designed to be very flexible to simulate complex quantum chemistry potential energy surface (PES). In general, MLFs have poor transferability, and hence a very large trainset is required to span all the target feature space to get a reliable MLF. This procedure becomes more troublesome when the PES is complicated, with a large number of degrees of freedom, in which building a large database is inevitable and very expensive, especially when accurate but costly exchange-correlation functionals have to be used. In this manuscript, we exploit a high dimensional neural network potential (HDNNP) on Pt clusters of size 6 to 20 as one example. Our standard level of energy calculation is DFT GGA (PBE) using a plane wave basis set. We introduce an approximate but fast level with the PBE functional and a minimal atomic orbital basis set, then a more accurate but expensive level, using a hybrid functional or non-local vdw functional and a plane wave basis set, is reliably predicted by learning the difference with HDNNP. The results show that such a differential approach (named ΔHDNNP) can deliver very accurate predictions (error < 10

meV/atom) in reference to converged basis set energies as well as more accurate but expensive xc functional. The overall speedup can be as large as 900 for 20 atom Pt cluster. More importantly, ΔHDNNP shows much better transferability due to the intrinsic smoothness of delta potential energy surface, and accordingly one can use much smaller trainset data to obtain better accuracy than the conventional HDNNP. A multi-layer ΔHDNNP is thus proposed to obtain very accurate predictions versus expensive non-local vdW functional calculations in which the required trainset is further reduced. The approach can be easily generalized to any other machine learning methods and opens a path to study the structure and dynamics of Pt clusters and nanoparticles.

Keywords: neural network, delta neural network, metal clusters, platinum

1 Introduction

The determination of the potential energy surface (PES) for molecular or solid state systems is fundamental for the theoretical studies of structure, dynamics and chemical reactions. Its description is however challenging for systems with a high number of degrees of freedom and complex interactions, and hence fast and reliable methods to evaluate the PES are under high demand. Nowadays, several options are available to evaluate the PES depending on different applications. Empirical force fields are very popular in simulating large biological systems, in which there is no bond breaking/forming events and thus harmonic approximations are good enough for simulating the PES. Empirical force fields are cheap and capable of biological simulations with as many as 10^6 atoms.[1] However, it is still a big challenge to develop accurate empirical force field for metallic systems, especially when heterogeneous surface reactions are involved[2-3]. Metallic systems generate complex inter-atomic interactions which are difficult to be simplified as additive models. Instead, density functional theory (DFT) calculations have become the standard approach to investigate metallic systems in the past decades because of the good accuracy while being more computationally efficient than wave function based methods. Nevertheless, even equipped with most advanced computers, current DFT calculations with semi-local exchange correlation functionals are only able to treat systems with several hundreds of atoms for dynamics within a few picoseconds. In many simulations, more expensive exchange-correlation functionals are required for accurate results, thus DFT calculations are even more CPU extensive, rending difficult long atomistic simulations or including large degrees of freedom.

Recently, Machine Learning force fields (MLF) approach have emerged as another promising method for obtaining DFT-level accuracy PES with orders of magnitude smaller CPU costs.[4-12] MLFs are purely data driven methods and they predict the unknown configurations using knowledge learned from existing references (trainset). Once the trainset is well prepared and the ML models are well trained, the predictions of MLFs are very close to the references methods. The reference method is density functional theory (DFT) in most applications. In the literature, there are several popular MLFs, like the high dimensional neural network potential (HDNNP)[9-10, 13-21] and the Gaussian approximation potentials (GAP)[22].

One of the main drawbacks of MLFs is that they are generally poor at generalization. This means that the unknown structure must be qualitatively similar as some structures in the trainset. Otherwise, the models are nearly guaranteed to provide wrong predictions. One intrinsic reason is that MLFs do not follow any physical laws, which governs the asymptotic phenomenon of the PES. Therefore, MLFs give accurate prediction only if the new configuration is very close to the structures in the trainset (in feature space). Hence, one has to build a trainset which covers nearly all the configuration space in order to get reliable predictions. Building a complete trainset is not an easy job because it is very difficult to know a priori which part of configuration space is missing in the trainset. Some methods have focused on detecting the 'extrapolated space' automatically.[23] Given those methods work ideally, one has to stop the simulation again and again to redo DFT calculations and to retrain the neural network potentials. Inevitably, one has to exploit a "trial and error" approach to test and extend the trainset, so that building and improving a trainset becomes a burdensome and time-consuming task because of the insufficient transferability of MLFs.

One straightforward way to circumvent this problem is to use an approximate method, like semi-empirical methods (for example, tight binding DFT) to teach the MLFs with prior-knowledge and then use the MLF approach to only train the difference between the approximate method and the reference method. This approach is also called Δ-ML method. For example, Ramakrishnan et al. used an approximate quantum chemical method (PM7) to calculate the atomization energy of organic molecules and then used a Δ-ML approach to account for thermodynamic corrections as well as high level electron correlation effects.[7] In Ramakrishnan's study, the machine learning method is only used for correcting the configurational and compositional space, while the conformational space (i.e. PES) is not discussed. One possible limitation is associated to the choice

of descriptors and they used the sorted Coulomb matrix, which is usually employed for learning the atomization energies of molecule in the ground state geometry rather than learning the PES. A similar approach was also investigated by Balabin et al. [24-25] and Xu et al. [26-27] to improve the prediction of atomization energies for organic molecules.

Another approach using Δ-ML was demonstrated by Lin S. et al. [28-29] They exploited a revised high dimensional neural network potential (HDNNP) to incorporate the reaction coordinate in a sub-neural network. A low-level semi-empirical QM/MM molecular dynamics simulations is performed to collect the data and the final potential of mean force (PMF) can be obtained by a reweighting procedure. This work however exploits semi-empirical quantum method which are not applicable to metallic system and on the other hand, the isomerization dynamics of metal clusters does not have well-defined reaction coordinates. In order to investigate metal clusters, the exploited MLFs have to be accurate for the whole PES in any hyper coordinate, not only one reaction coordinate.

In this contribution, we will introduce a very simple but flexible scheme called hierarchical delta neural network method, which exploits one or more layers of HDNNP to account for the target differences between low-level and high-level calculations. This method is notated as Δ_sHDNNP, in which **s** indicates the number of Δ layers. The differences between calculation levels can be linked to the choice of an inaccurate, but small basis set or to the selection of different xc functionals. In the simplest case of only one layer (s=1), we exploit DFT with single zeta basis set as low level approximation and ΔHDNNP is used for correcting both the differences from basis sets as well as xc-functionals. HDNNP is used because the framework of HDNNP is flexible and in general, it can simulate any complex PES (or ΔPES). Although DFT with single zeta atomic basis set increases the computational cost in productive simulations, we will demonstrate that Δ_sHDNNP delivers much better transferability and requires significantly smaller trainset due to the intrinsic smoothness of ΔPES. Δ_s HDNNP requires a train set 10 times smaller than that of a direct HDNNP while retaining better accuracy. What is more important, it is straightforward to generalize Δ_s HDNNP to s > 1., i.e. a multi-layer delta neural network approach when a proper auxiliary level DFT is used. A multilayer Δ_s HDNNP can be very useful for the situation where an expensive xc-functional is necessary and it is hence costly to generate large reference database.

2 Method and calculation details:

2.1 High dimensional neural network potential (HDNNP)

HDNNP has been a very popular machine learning force field method in the past decade. It was first introduced by Behler et al.[9-10, 16-18, 20, 30] The essential idea of HDNNP is that the total energy of a quantum chemical system E_t can be calculated by summing up energies of individual atoms E_i.

$$E_t = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} NN(X_i^{env}) \quad (1)$$

In equation (1), N is the total number of atoms and E_i is the energy of atom i. E_i is determined by the chemical environment of atom i and the latter is characterized by the feature vector **X**, which is constructed with symmetry functions in the original paper.[20] Besides symmetry functions, other types of descriptors have also been invented to transform the chemical environment into the feature space including Chebyshev polynomials[31], Zernike descriptor and a bispectrum descriptor[21]. A neural network (NN) is exploited to discover the relation between the feature vector **X** and the atomic energy E_i. Descriptors are designed to be rotational invariant and the HDNNP uses the same NN model for each atom of one element type to fulfill the permutation symmetry. Therefore, the PES from HDNNP follows basic symmetry requirements of first-principles derived PES. In principle, HDNNP is also cable of simulating chemical system with variable number of atoms and compositions. Those advantages make HDNNP very useful in theoretical simulations.

In this manuscript, we exploit two different kinds of descriptors to compute the feature vector **X**. The first one is a set of symmetry functions following the original paper of Behler et al.[16, 20] The symmetry functions consist in a set of radial and angular functions to describe the environment of one atom. The radial part is shown in Equation (2):

$$G_i^2 = \sum_{j=1}^{N} e^{-\eta\left(r_{ij}-R_s\right)^2} f_c\left(r_{ij}\right) \quad (2)$$

$f_c(r_{ij})$ is the cutoff function to ensure that the function will vanish beyond cutoff R_c ($R_c = 6.5$ Å for symmetry functions in this study). r_{ij} is the distance between center atom i and neighborhood

atom j, R_s and η are chosen parameters. The angular part takes into account the bond angles between triplets of atoms i, j, k, in which the θ_{ijk} is the angle between bonds ij and ik.

$$G_i^4 = 2^{1-\zeta} \sum_{j,k} \left(1 + \lambda cos\theta_{ijk}\right)^{\zeta} e^{-\eta\left(r_{ij}^2 + r_{jk}^2 + r_{ki}^2\right)} f_c\left(r_{ij}\right) f_c\left(r_{kj}\right) f_c\left(r_{ik}\right) \quad (3)$$

In equation (3), ζ, λ and η are also chosen parameters. In order to find a suitable set of parameters, we systematically increased the number of symmetry functions following the procedures proposed recently by Imbalzano et al.[32]. The details for determining the symmetry function parameters in each case are described in the Supporting Information (section 1.2).

The second type of descriptor is called Chebyshev polynomials, as recently proposed by Artrith et al.[31] Chebyshev polynomials also transform the radial distribution function (RDF) and angular distribution function (ADF) to a feature space vector **X**. Essentially, the ADF function and RDF function are projected on the basis set functions of Chebyshev polynomials and the expansion coefficients are used as the descriptors for atom's environments. For more details of the method implementation, we refer the supporting information of the original paper from Artrith et al[31]. One advantage of Chebyshev polynomials is that the size of **X** can be systematically enlarged by increasing the order of the Chebyshev polynomial.

The training and validation of HDNNP throughout this paper uses the Atomic Energy Network (aenet) package.[15, 31, 33] The architecture of the neural network is notated as X-(m×n)-1. X is the input layer size (length of **X**), m is the number of hidden layers and n is the number of nodes per layer. Therefore, the numbers of nodes in each hidden layer is the same (equal to n). Totally 9 different architectures are first explored for each trainset, in which m uses 2, 3 or 4 and n uses 5, 15 or 30. The training results are given by an early-stop scheme (see Figure S1) or from the last iteration of the training. Because neural network is a non-convex function, the optimization algorithm is not guaranteed to locate the global minimum solution. To circumvent the uncertainty in the optimizations, we performed three independent training for each case. The standard deviation among the three training results are reported to indicate the repeatability of the training results, though evaluation of the standard deviation between different trainings are not necessary in productive calculations, one can choose the best model which shows the smallest errors.

2.2 DFT methods for building database

The first reference database used in this manuscript consists of small Pt clusters whose sizes range from 6 to 20 atoms. Initial coordinates are randomly generated and then extended by local optimization steps or short MD steps. Only compact clusters are collected and the total size of the database is 6402. Energies and forces are computed by DFT with The Vienna Ab initio simulation package (VASP).[34-37] The cutoff for plane waves is 250 eV and the Perdew–Burke-Ernzerhof (PBE) functional[38-39] is exploited to describe electronic exchange and correlation. Only gamma point (1×1×1 **k**-mesh) is used for sampling Brillouin zone. DFT calculations are non-spin polarized throughout this manuscript since the force field is an approximation of the non-spinpolarized DFT accuracy in the best sense. The database or DFT calculation method are labeled with the notation: *functional (package-basis)*, therefore, the first database is referred as PBE(VASP-PW).

On the other hand, we used DFT with small basis sets as low level approximations. The methods are labeled as PBE(GPAW-SZ) or PBE(CP2K-SZV) depending on the software and basis set names.

We also built two reference database with more accurate functionals called TPSSh(VASP-PW) and optPBE-vdW(VASP-PW), which exploits the more expensive non local functionals (optPBE-vdw) or hybrid functionals (TPSSh)[40-41]. These are two different methods showing improved catalytic predictions compared with the PBE functional.[42-44] TPSSh(VASP-PW) and optPBE-vdW(VASP-PW) databases are both smaller in size (2362 structures in total) and elements are randomly chosen from the PBE(VASP-PW) database.

3 Results and discussions

3.1 Direct training by HDNNP

We first investigated the conventional way to train a HDNNP for the Pt cluster database we built. Here we refer it as direct HDNNP since we directly exploited the reference energies. The ultimate goal of training HDNNP is to use a small trainset to obtain a HDNNP with good transferability and accuracy. Meanwhile, it would be of great advantage that the training procedures rely little on the users' skills or experience. Generally, there are three aspects influencing the quality of the HDNNP: 1) the choice of feature vectors, i.e. the descriptors **X**. 2) the architecture of the hidden

layer structures 3) the size of reference structures. For the input layer, we investigated two types of descriptors, with symmetry functions or Chebyshev polynomials.

3.1.1 Influence of input layer.

The HDNNP is first trained against the reference database PBE(VASP-PW). We systematically increased the size of the input layer with the methods presented in 2.1 and in Supporting Information Section S1. The architecture of the HDNNP is kept as **X-(*2*×*30*)-1** in this part and we will demonstrate later that this choice shows the best accuracy among the selected 9 architectures. The database is randomly split into a trainset (90%) and a control set (10%). The influence of different input layer types and sizes is summarized in Figure 1. The random splitting is commonly used in the literature as a way to split the database as trainset and control set in order to avoid overfitting. We also examined the distribution of fingerprints in Figure S11 to ensure that the ranges of fingerprint values are similar in trainset and control set. We present the quality of the descriptors as a function of a hyper parameter, the size of the descriptors. The relationship between the size of the descriptors and their parameters is discussed in section S1. For a particular size of descriptors, we may have different sets of parameters and we will only consider the set giving the smallest error in Figure 1 and Figure 4 (see the details in Table S1).

Figure 1(a) shows that increasing the size of the input layer (using more symmetry function as descriptors) from 20 to about 46 efficiently decreases the RMSE on the trainset. However, when the size of the input layer goes beyond 46, the RMSE of the trainset does not change significantly anymore. If we now look at the control set, the RMSE does not decrease anymore beyond size 32, remaining almost the same between 32 and 46 and slightly increasing by 1 meV/atom when the input layer size goes beyond 46. Those results show that the symmetry function set with size 46 gives the optimal performance and also remains computationally efficient. We also compared two different ways for selecting the trained potentials. The first way uses the early-stop scheme (as shown in Figure S1) and the second way uses the final iteration (which is the 5000th iteration in all of examples in Figure 1 and Figure 4). The two different methods show very similar information about the quality of the descriptors. Similar numerical experiments are also conducted with Chebyshev polynomial descriptors. The performance of the Chebyshev polynomials are overall very similar with that of symmetry functions, though the calculated RMSEs using Chebyshev polynomials are slightly larger (by about 1~2 meV/atom) than that of the symmetry functions. This

very small difference between the two types of descriptors may come from the choice of parameters.

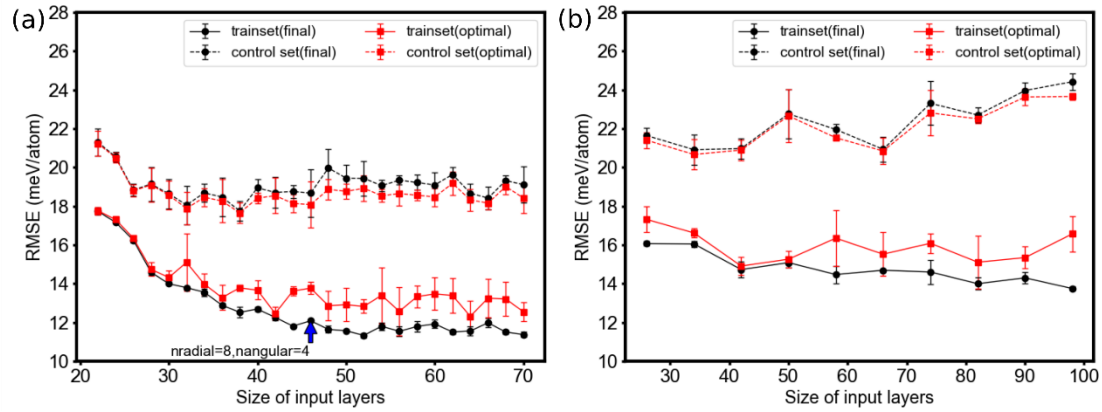

Figure 1. Direct training results of HDNNP using the PBE(VASP-PW) database. The RMSEs (meV/atom) of trainset and control set are given for different sizes of the input layer. The architecture of the neural network is kept as X-(2×30)-1, where the x axis is the size of the input layer (X). The y axis is the RMSE averaged on three independent trainings and the error bars indicate the variation between them. Two types of input layers are considered, including (a) symmetry functions (Symfunc) and (b) Chebyshev polynomials (Chebyshev). In each subfigure, the final sets (black lines) indicate the RMSE at the 5000th iteration step and the optimal sets (red lines) indicate the RMSE obtained from the early stop scheme.

3.1.2 Influence of hidden layer architectures

In order to investigate the influence of hidden layer architectures on the performance of HDNNP, we carried out trainings with fixed symmetry function parameters as input layers (X=46). The results are shown in Figure 2 and the architectures of the neural network are 46-(m×n)-1. The results show that m does not have a significant impact on the performance as long as n is the same. Increasing m from 2 to 4, the root mean square errors (RMSEs) of the test and train sets are very similar. In contrast, n has very obvious influence on the performance. Figure 2 shows that by increasing n from 5, 15 to 30, the RMSEs for the control sets are 22, 18, 17 (meV/atom) respectively. Increasing the number of nodes from 15 to 30 does not significantly improve the RMSEs for the control set, though RMSEs for the trainset can be reduced by 3 meV/atom. The significantly larger differences between RMSE for the control set and for trainset when hidden nodes is more than 15 nodes /layer indicate that the trained model suffers from overfitting in some extent.

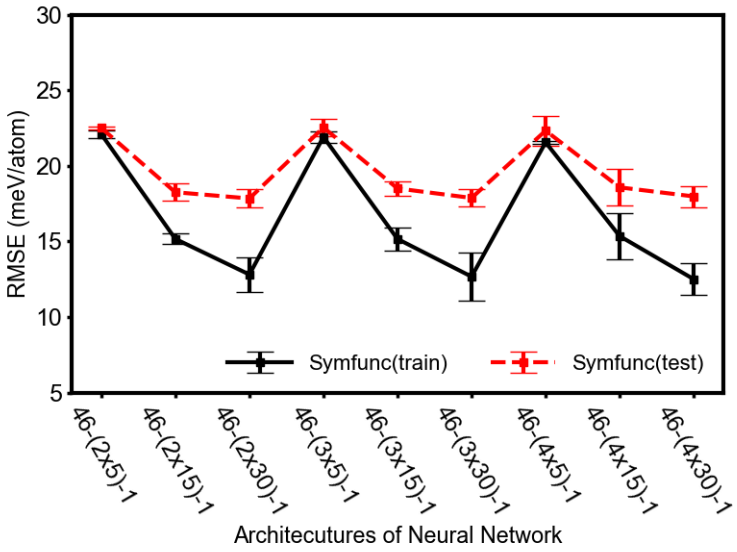

Figure 2 Effects of different architectures on the training errors. The trainset is PBE(VASP-PBE). The input layer size is 46 using symmetry function descriptors. The y axis is the averaged RMSE from three independent trainings and the error bars indicate the variation between them.

3.1.3 Influence of trainset size

In section 3.3.1 and 3.1.2, the PBE(VASP-PW) is randomly split as trainset and control set, in which the trainset contains 90% of the database. It will be very important to know whether it is possible to reduce the number of structures in the trainset without impacting the RMSE for the control set. Therefore, we conducted more trainings by gradually reducing the trainset size from 90% to 10% of the database and all the other structures in the database are used as control set. The results are shown in Figure 3. Obviously, with smaller trainset, the RMSE for the control set gradually increases from 17 meV/atom to about 28 meV/atom. Hence one cannot safely use smaller trainset in HDNNP.

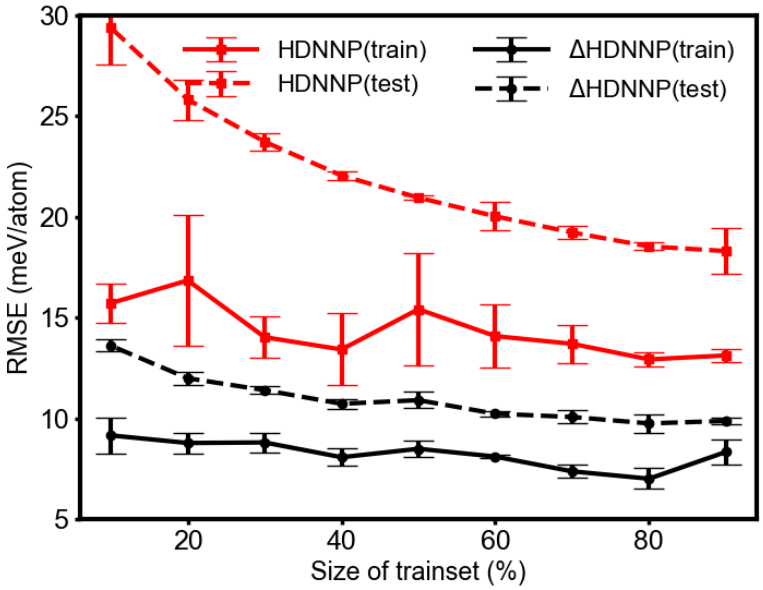

Figure 3. Influence of trainset size on HDNNP and Δ_1 HDNNP errors. The total size of the data base is kept fixed (N=6402) in each training. The percentages of the trainset are different and range from 10 % to 90 %. All other structures in the database not used for the trainset are used in the control set. The neural network architecture is 46-(2x30)-1 with symmetry functions as descriptors for the input layer.

Section 3.1.1 and section 3.1.2 demonstrate the difficulty in training a HDNNP: it is impossible to know in advance the optimal neural network architecture (like the input layer size and hidden layer architecture). Although larger size of input layer is more capable of differentiating the local atomic environments in principle, using large input layers does not always improve the performance. Using very large hidden layer architecture would result in a very flexible neural network but also in a risk of over-fitting. Meanwhile, the trainset size has to be very large in order to retain an RMSE below 20 meV/atom for the control set.

It will not be a surprise that the performance of HDNNP can be continually improved by constantly following trial-and-error procedures via using larger trainset, using more complicated neural network architectures and descriptors. Nevertheless, because of the black-box nature of neural network learning algorithms, the training procedures will be very time-consuming or the size of the required database might become inaccessible before achieving an ideal trainset. The reference database collection and training procedures might not be worthy anymore compared with the further productive simulations. Obviously, a new approach, which can reduce the requirement to the trainset and be less dependent on the training experience, would be very useful.

3.2 Delta training:

In this part, we exploit faster density functional theory calculations which uses single zeta basis sets and PBE functional (named as PBE(GPAW-SZ)). PBE(VASP-PW) is regarded as standard accuracy for Pt clusters, while single zeta basis sets are generally not accurate enough but are more efficiently in computation (Figure S2 shows the comparison of accuracy and timings are shown later). The reference database PBE(VASP-PW) is recalculated by PBE(GPAW-SZ) and the energies from PBE(GPAW-SZ) is then subtracted from the reference energies (PBE(VASP-PW)) to build the Δ database, which consists of the basis set effect between SZ atomic orbitals and plane waves. In this work, we subtracted directly the energies from different packages and we did not scale the energies before preparing the delta database. The aenet package will automatically normalize and shift the reference energies to [-1.0, 1.0] before training. In this particular example, the Δ database also includes the very small numerical errors from different code implementations, and one can expect smaller RMSE of this Δ approach if the code implementation errors could be excluded. By all means, the small error in code implementation does not impact our conclusion.

The performance of Δ_1 HDNNP is shown in Figure 4 and Figure 5. Figure 4 shows a very similar message compared to Figure 1. With a similar input layer size, symmetry functions slightly outperform the Chebyshev polynomials by an averaged error reduction of 1.0 meV/atom. Increasing the input layer size does not necessarily lower the RMSE of control set. The effects of the hidden layer architectures are also demonstrated in Figure 5 and the trend is the same as Figure 2. First, the number of hidden layers does not help to significantly reduce the error. Second, with increasing the number of nodes in each layer, the RMSE of trainset keep decreasing, but the overfitting becomes more significant.

The major difference between HDNNP and Δ_1 HDNNP is that the RMSEs on the trainset and the control set are both significantly reduced from 14/18 to 8/10 (RMSE (control set)/RMSE(trainset) when X=46; units meV/atom). Δ_1 HDNNP reduces the error of HDNNP by about half. This is a notable improvement for practical applications. For example, the clusters used in the current manuscript consist of different sizes and the maximum size is 20. HDNNP shows a prediction error of 18 meV/atom in the control set and it means the error on total energy could be as large as 0.36 eV. Δ_1 HDNNP only introduces an error of 0.20 eV in the total energy instead. It is known that the number of Pt cluster isomers within an energy range increases exponentially[45], a small

increase in the error of the total energy will bring significant uncertainty to the stability order computed by HDNNP.

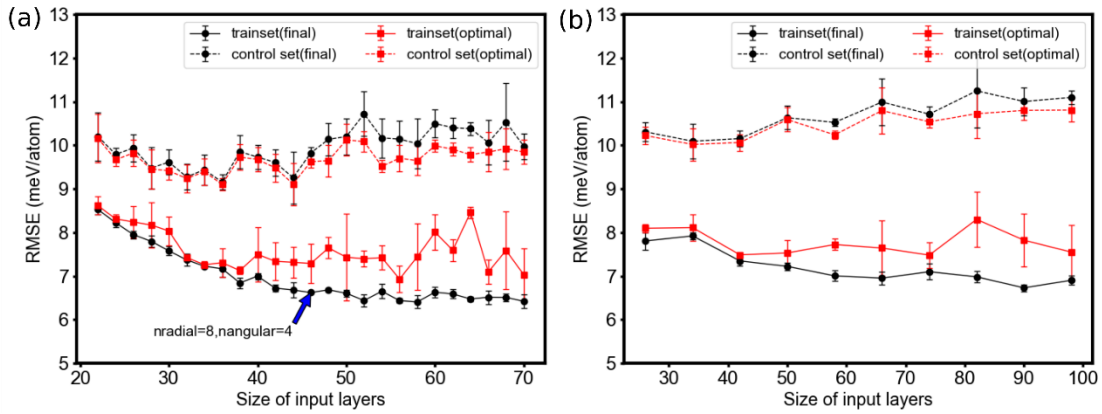

Figure 4 The RMSEs on the trainset and control set with different input layers. In all the Δ_1 HDNNP, the hidden layers consist of 2 layers and each layer contains 30 nodes, and randomly 10 percent of the database is used as control set. PBE(GPAW-SZ) is exploited as the low level approximation method. Two different types of descriptors are used, including (a) symmetry functions (Symfunc) and (b) Chebyshev polynomials (Chebyshev). In each subfigure, the final sets (black lines) indicate the RMSE at the 5000th iteration step and the optimal sets (red lines) indicate the RMSE obtained from the early stop scheme.

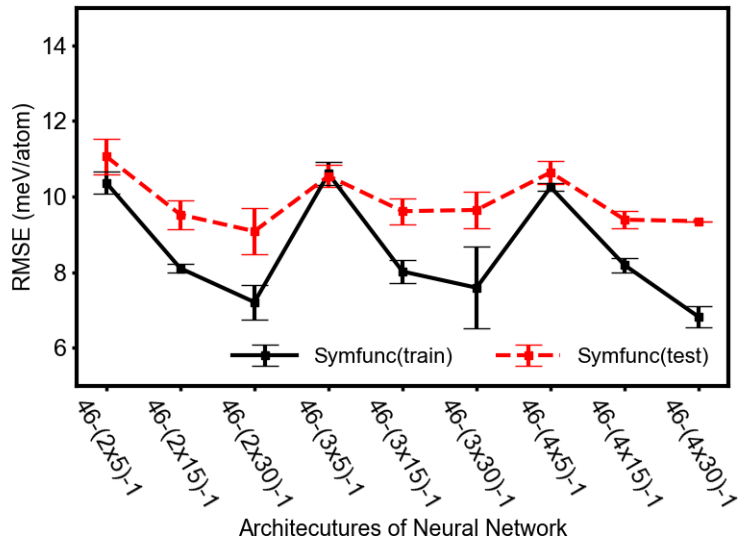

Figure 5 RMSEs on the trainset and control set with different hidden layer architectures using the Δ_1 HDNNP approach. The input layer is kept fix as 46, and randomly 10 percent of the database is used as control set. PBE(GPAW-SZ) is exploited as the low level approximation method.

In addition, Δ_1 HDNNP requires a significantly smaller trainset and in the meanwhile, it always delivers a very small RMSE versus the reference database. Figure 3 clearly demonstrates that

accuracy of Δ_1 HDNNP is better than that of HDNNP (RMSE reduced by a half for the control set) no matter what is the size of the trainset. In the extreme case, only 10 % of the database is used as trainset for Δ_1 HDNNP and the performance is still markedly better than that for HDNNP using 90 % of structures as trainset. When repeating the training three times, all of the Δ_1 HDNNP training performs smoothly without any manipulations of the parameters. One possible concern is that Δ_1 HDNNP requires an extra computational cost to calculate the low level approximation, which is not required in original HDNNP. Here, Figure 3 underlines that the extra cost to calculate PBE(GPAW-SZ) can be compensated by the better transferability of Δ_1 HDNNP and thus Δ_1 HDNNP requires a much smaller trainset database.

We should point out that the advantages of Δ_1 HDNNP versus HDNNP do not result from the code implementation or using a special parameterization of the single zeta basis set, instead it is intrinsic that Δ database is easier to be learned compared with the original PBE(VASP-PW) one. To prove this conclusion, we also use a different DFT package, CP2K,[46] which comes with a different parametrization of single zeta basis set (SZV in the molecularly optimized basis functions[47]) to calculate the Δ database. Then we re-calculate the Δ database and carry out the training with similar procedures as in Figure 3. The results are shown in Figure S4, and the performance of the Δ_1 HDNNP approach is very similar (within 1 meV/atom) when exploiting different codes for low-accuracy energies. Another comment relates to the distribution of averaged atomic energies, shown in Figure S10. It is clear that the ranges of the reference energies and the delta energies (from the difference between PBE(VASP-PW) and PBE(GPAW-SZ)) are almost the same, though the FWHM (full width at half maximum half-width) of direct energies' distribution is slightly larger than that of delta energies. The small difference in FWHM of the energy distribution is not able to explain the different training performance. The more significant difference is that direct energies' distribution is bimodal unlike the unimodal distribution of the delta energies, which is a clue of a simpler (smoother) target function in Δ_1 HDNNP.

3.3 Smoothness of the Δ database

The HDNNP learns to predict the quantum chemistry energy of structures by their similarity in feature space. During the training, the neural network attempts to learn the mapping $f: R^N \rightarrow E$, in which R^N corresponds to the geometry features (either Cartesian coordinate space or descriptor space) and E is the energy. The smoothness of the target function f is very important for the

transferability of the trained model. In general, one would think that the poor transferability of HDNNP comes from the fact that HDNNP fails to extrapolate from known feature space. However, the concept of 'extrapolation' is blurry in high dimensional space. Let us assume for example that we conduct a machine learning task with a feature space of dimension 3 (the learned model is f') and we would like to predict the property at the origin ($f'(x = \mathbf{0})$). If all the feature vectors of the reference data are far from the origin and located on or beyond the sphere ($r = |\boldsymbol{R}|$), it will be ambiguous whether the prediction $f'(x = \mathbf{0})$ is 'extrapolated' or 'interpolated'. The prediction can either be extrapolated if the correlation distance is smaller than $|\boldsymbol{R}|$ or interpolated if the correlation distance is larger than $|\boldsymbol{R}|$. This scenario is very common in conducting machine learning in a high dimensional feature space and is referred to as the *curse of dimensionality*. In a high dimensional space with evenly distributed references, all data points are far away from each other. There is not a clear way to distinguish the 'extrapolated' or 'interpolated' region. It is obvious that the 'smoothness' of the function is relevant with the correlation distance. If the correlation distance is large, the function is 'smoother' and easier to be generalized. Otherwise, the function is rough and more difficult to be generalized. Therefore, the correlation distance, or the covariance matrix between the references and the new data points are more realistic concepts than the range of feature vectors.

Of course, the previous example is only a simplified illustration, and not a rigorous proof of the effect of the target function's smoothness on the learnability from a machine learning method. A rigorous mathematical proof on the complexity/smoothness of the target function f is beyond the scope of the current contribution. However, we can still obtain some clues on the reason why learning the target function $f: R^N \rightarrow \Delta E$ (mapping geometry to delta energies) is easier than learning the target function $f: R^N \rightarrow E$. One approach is to show that the *correlation distance* is larger for the delta database, i.e. that structures who are close in feature space provide similar delta energies, but not similar direct energies.

First, please note that fingerprints (input layer of neural network) are normally not the optimal representation for atomic energies, so that the Euclidean distance in the original fingerprint space might not be the correct measure to evaluate the proximity between two structures. This is inferred by the PCA analysis on the input layer of the neural network shown in Figure S12(a) and (c), where there is no correlation between energy and principal axis. A better representation (transformation

of the original feature vector) is required to investigate the correlation distance between structures in feature space. In order to show that the ΔHDNNP consists to learn a smoother function ($f: R^N \rightarrow \Delta E$) than that of HDNNP $f: R^N \rightarrow E$, we plotted the t-Distributed Stochastic Neighbor Embedding (t-SNE) analysis of the structural fingerprints instead, and each atom is colored by the predicted atomic energies.[48-49] The t-SNE analysis simulates the joint probability to find neighboring points in the high dimensional feature space through their distance in a 2-d layout. Therefore, we expect that, in the context of t-SNE transformed features, similar atoms should have a similar color, if the function f is smooth. Apparently, Figure 6(b) has a better energy separation than Figure 6(a) implying a larger correlation distance in $f: R^N \rightarrow \Delta E$. In other words, the better color separation in Figure 6(b) indicates that the learned function $f': R^N \rightarrow \Delta E$ is smoother than $f': R^N \rightarrow E$ (prime means the trained model rather than the true one). Considering that neural network training is a process to mimic the true function f with the learned model f', we can expect the true function $f: R^N \rightarrow \Delta E$ is also smoother than $f: R^N \rightarrow E$. Hence, ΔHDNNP is targeting at a smoother function than HDNNP, and therefore it shows better transferability.

On the other hand, the neural network is also a way to transform the input layers (initial descriptors) into different better representations and to store them in hidden layers. Therefore, the last hidden layer is an optimal representation to predict the atomic energies proposed by the neural network. This was previously investigated by Cubuk et al.[50] Following a similar approach, we applied PCA to the representation of the atomic energy E_i from the last hidden layer of the HDNNP. The results are shown in Figure S12(b) and (d). One can see that after the non-linear transformation with the neural network, the output of the last hidden layer provides a much better representation of the atomic energies compared with the input layer. Finally, after the neural network transformation, the PCA analysis on the delta database using the output of the last hidden layer shows a better correlation between atomic energies and first principle axis (Figure S12(d)) than in the case of the direct database (Figure S12(b)), implying that the neural network provides a simpler representation in the case of the delta database training. This is also a sign that the target function $f: R^N \rightarrow dE$ is simpler to be generalized.

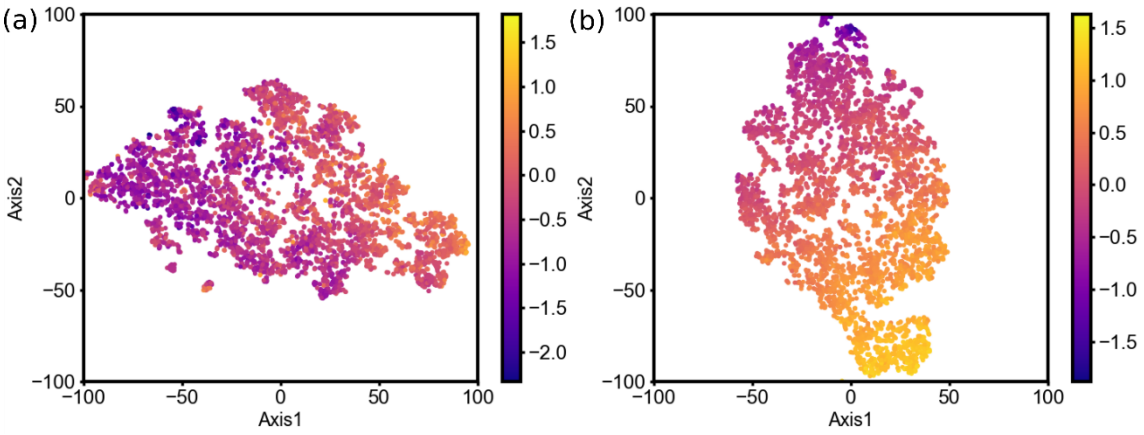

Figure 6 t-SNE analysis with symmetry function values as inputs and learned atomic energies as color code (a) using HDNNP learned atomic energies as color code. (b) using ΔHDNNP learned atomic energies as color code. Both of them use 90% of the database as trainset and the architecture of the neural network potential is 46-(2x30)-1. Only 7% of the atoms (randomly selected) are shown in the figures for clarity purposes.

3.4Improvements on the force accuracy.

In order to exploit MLs potential for structure optimizations, the accuracy of the predicted force is also important besides the energies. Although the HDNNP trainings only use energies, it also produces forces associated with the trained PES. Figure 7 shows the errors of force prediction by HDNNP and Δ_1 HDNNP respectively. It is clear that Δ_1 HDNNP reduces the force error of HDNNP from 0.96 eV/Ang to 0.47 eV/Ang in the case where 50% of the database is used as trainset (other results are in Figure S6 and Figure S7). The large force errors shown in Figure 7 may result from two parts. First, force is more sensitive to the accuracy of energy prediction as well as the transferability of the neural network models, the errors on force prediction are normally larger in absolute values. Second, the reference method PBE(VASP-PW) uses only plane waves up to 250 eV and this parameter provides good energy but poor force quality. Therefore, the noises in the reference database adds into the comparison difference. However, it is clear that Δ_1 HDNNP still provides much better forces compared with HDNNP with the same size of the trainset. Since the force is the gradient of the potential, if the potential energy surface is very rough, it will be very difficult to predict the forces with very sparse reference data points. In contrast, it will be easier to predict the forces when the potential energy surface is smoother. Therefore, the better force prediction with Δ_1 HDNNP provides another indication on the smoothness of delta energies.

Of course, It can be expected that Δ_1 HDNNP can be improved when we increase the reference force quality and database size.

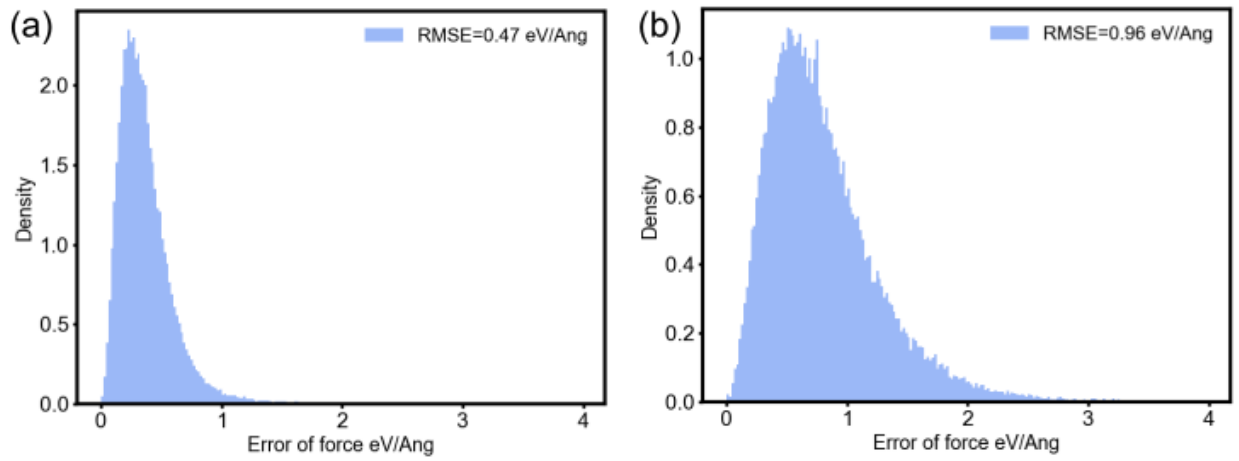

Figure 7 Comparison of the force prediction from Δ_1HDNNP (a) and HDNNP (b). The architecture of the neural network is 46-(2x30)-1 with symmetry functions as input descriptor. 50 % of the database is used as trainset and the rest is used as control set. The figure shows the distribution of the force error $\left|\vec{F}_{NN} - \vec{F}_{ref}\right|$, where the reference method is PBE(VASP-PW).

3.5 Transferability of ΔHDNNP in Out-of-Sample Data

To further investigate the transferability of Δ_1 HDNNP, new structures which are not included in the PBE(VASP-PW) database are used. The new structures are generated by following procedures: 15 structures are randomly selected from the PBE(VASP-PW) database. Then, constant temperature MD simulations are carried out starting from those 15 structures respectively using the built Δ_1 HDNNP, in which temperatures are chosen as 300 K and each MD simulation runs 100 fs with steps of 1.0 fs. Then, the energy of each structure in the MD trajectory is re-evaluated by the reference method PBE(VASP-PW). The same procedure is also conducted with the HDNNP for comparison. The results are shown in Figure 8. In the trajectories generated by HDNNP, even if some trajectories start with rather small errors (smaller than 10 meV/atom), they quickly diverge from the reference energies in 20~30 fs. On the other hand, errors along the Δ_1 HDNNP generated trajectories do not increase (in average) during the MD simulations. To explore the reason of the better transferability of Δ_1 HDNNP in Figure 8, we examined the ranges of input symmetry functions for the structures generated by MD and compared them with that of the trainset data (Figure S13). The results show that the fingerprint ranges of MD generated structures are still

within that of trainset structures, no matter they are generated by HDNNP or Δ_1 HDNNP. Hence, though the direct HDNNP extrapolates poorly on those out-of-sample data and results in a large RMSE, one cannot identify this by simply examining the ranges of fingerprint values.

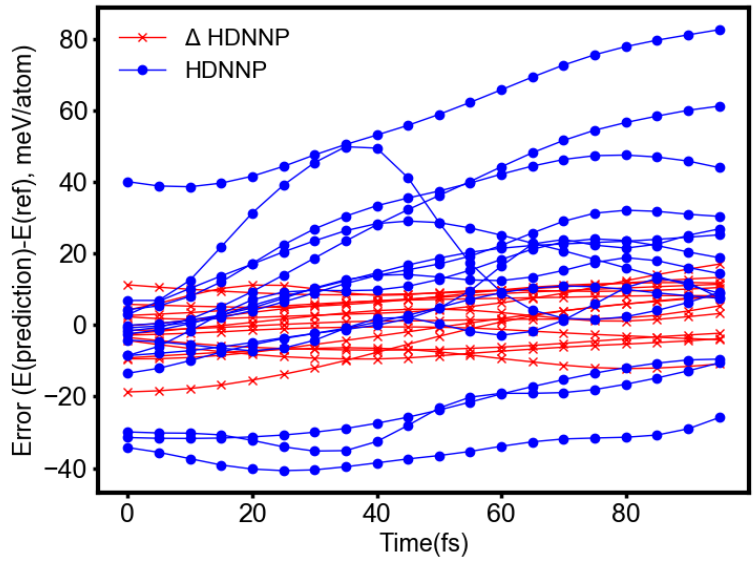

Figure 8. Errors of energies in molecular dynamics trajectories, starting from 15 structures from the trainset. The blue lines with round dots show the errors (per-atom, units are meV) from the HDNNP generated trajectories; red lines with crosses show the errors from the Δ_1 HDNNP generated trajectories.

A second numerical experiment to demonstrate the good transferability of Δ_1HDNNP uses Pt_{25} and Pt_{55} clusters. We remind that the largest size of the Pt clusters in the trainset is 20 atoms. To investigate whether the Δ_1HDNNP is able to describe clusters larger than 20, we selected 60 Pt_{25} and 60 Pt_{55} clusters from a long MD simulation with EAM potential and then evaluated the error between the two different NN potentials and the DFT reference. The results are shown in Figure 9. Because Pt_{25} and Pt_{55} clusters are completely excluded from the original trainset database, one can expect larger prediction errors from either HDNNP or Δ_1HDNNP. However, results show that the Δ_1 HDNNP performs very differently from HDNNP. For Pt_{25} cluster, Δ_1 HDNNP shows an averaged absolute error about 28.2 meV/atom versus the reference, which is indeed larger than the error in the previous smaller clusters. However, Δ_1 HDNNP presents a constant and small offset from the reference energies, and the relative stability of different clusters is still rather reliably predicted: the relative error on the stability are just 10.8 meV/atom. On the other hand, the absolute error and relative errors for the HDNNP method are 195.2 meV/atom and 26.9 meV/atom, being hence significantly larger than that of ΔHDNNP. Therefore, ΔHDNNP still resembles the potential

energy surface of Pt_{25} and demonstrate very good transferability. The results of Pt_{55} clusters are even more remarkable, though they seem more challenging at first glance because Pt_{55} contains two complete shells completely excluded from the trainset. The results in Figure 9(b) show that the absolute error between Δ_1 HDNNP and DFT is 20.8 meV/atom, while the relative error of Δ_1 HDNNP only present 5.5 meV/atom. Hence, the relative energy among different clusters are still well predicted. In contrast, the direct HDNNP completely fails to describe the relative energies of different geometries. This example shows that Δ_1 HDNNP can reliably predict the relative energies of large clusters, although clusters of such sizes are not used in the trainset.

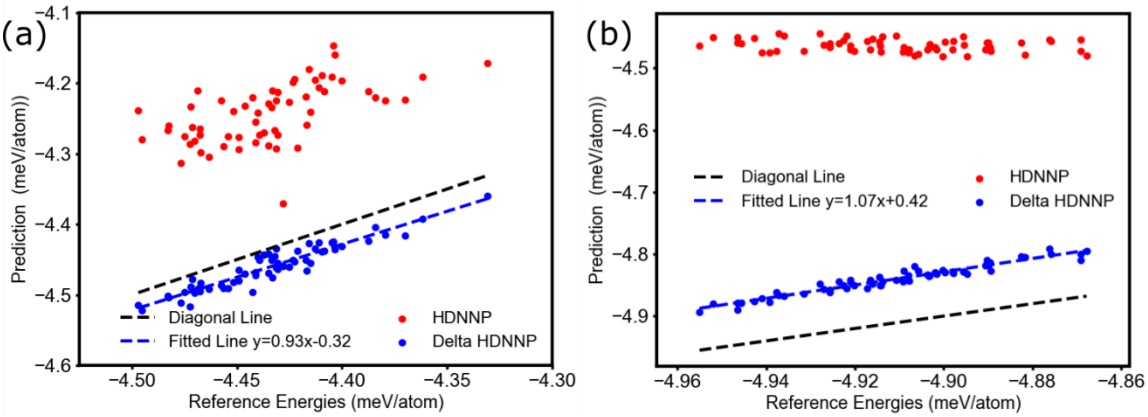

Figure 9 Comparison between the predicted per-atom energies from HDNNP (red dots) or ΔHDNNP (blue dots) and reference values (PBE(VASP-PW)) for (a) 60 structures of the Pt_{25} cluster (b) 60 structures of the Pt_{55} cluster. The black dashed line indicates the diagonal and the blue dash line is the fitted linear equation between the ΔHDNNP and PBE(VASP-PW) energies.

4 Using ΔHDNNP to go beyond GGA functionals.

Unlike empirical force field, MLFs always requires a train set of large size to get reliable predictions. Currently, DFT with GGA xc functional is commonly used as the reference method, which enables fast energy-force calculations. While, GGA functionals are reasonable cheap but not sufficiently accurate for many applications.[41, 51] Even if the trained MLFs are excellent to model the GGA reference, the MLFs potentials are still subject to the errors originating from the DFT-GGA calculations. One way to circumvent this bottleneck is to exploit more advanced xc functionals like non-local functional or hybrid functional, providing much better accuracy.[43-44, 52-56] Nevertheless, the non-local functional and hybrid functional are very expensive compared with GGA functional, and hence building the required large number of trainset structure may be inaccessible.

Our solution is to take advantage of the high transferability of Δ_1HDNNP. The Δ_1HDNNP approach has been demonstrated to be a very efficient way to simulate the energy difference caused by basis set inadequacy like between minimum single-zeta and plane wave basis sets in Figure 3. In addition, one can take one step further and use a Δ_1HDNNP approach to account for other aspects of electronic structure calculations, providing corrections for different xc functionals. For example, we built two databases similar to the PBE(VASP-PW) database but using the TPSSh or optPBE-vdW functionals, databases called TPSSh(VASP-PW) and optPBE-vdW(VASP-PW) respectively (N=2362). TPSSh is a hybrid version of meta-GGA TPSS functional, showing very good accuracy for Pt clusters.[57-58] optPBE-vdW is based on the non-local correlation functional from the Rutgers-Chalmers van der Waals Density Functional (vdW-DF) combined with an optimized PBE-like exchange functional.[59-60] It correctly describes adsorption properties of Pt.[44] Therefore, we take TPSSh and optPBE-vdW as examples to demonstrate the strength of Δ HDNNP.

First, we compared the energy difference between PBE and optPBE-vdW or TPSSh respectively and the results are shown in Figure S3. Both the optPBE-vdw and TPSSh functionals predict different stabilities of Pt clusters compared with PBE. OptPBE-vdW shows an average difference of 26.4 meV/atom, which is still very significant considering the size of the clusters. The TPSSh functionl shows more corrections[61] and the cohesive energy predicted by TPSSh not only shows a large systematical offset, and an average deviation from PBE as large as 162.2 meV/atom.

Then we exploit PBE(CP2K-SZ) as low level DFT method to train the Δ_1HDNNP targeting at the more expensive functional TPSSh or optPBE-vdw. The architecture of the HDNNP is 46-(2x30)-1 with symmetry functions as descriptors. The results are shown in Figure 10 and we can see that ΔHDNNP with PBE(CP2K-SZ) as low level DFT outperforms HDNNP for both TPSSh and optPBE-vdw functional. HDNNP shows control set error around 23 meV/atom for both functionals, even with 90% of the database as trainset. While, if PBE(CP2K-SZ) is used as low level DFT to train Δ HDNNP, the RMSE on the control set is significantly reduced. The reduction is more significant when the high-level xc functional is optPBE-vdW, in which the RMSE is decreased from 23 meV/atom to 10 meV/atom. In the case of TPSSh, the improvement is slightly smaller, which a reduction of the RMSE from 23 meV/atom to 15 meV/atom. The transferability is also improved. When only 10 % of database is used as trainset, the RMSE on control set

increases from 23 meV/atom to 35 meV/atom (or 37 meV/atom) in HDNNP, while the increasing is only 4 meV/atom (or 7 meV/atom) for ΔHDNNP in the case using optPBE-vdW (or TPSSh) as target xc functional.

Another numerical experiment uses PBE(VASP-PW) as low level DFT to train ΔHDNNP targeting at more expensive xc functionals. In this set, the energy difference only comes from the xc functionals. This approach is still useful considering the large CPU efficiency difference of different xc functionals (shown later). The results are also given in Figure 10. It is clear that the Δ database is even more easily trained with PBE(VASP-PW) as low level DFT. One can use only 10% of the expensive optPBE-vdW database to train the ΔHNNP reproducing the functional difference between PBE and optPBE-vdW with a RMSE of only 1.5 meV/atom (Figure 10(a)). In the case of the TPSSh functional, using PBE as a low level method for ΔHDNNP gives a RMSE on the control set reducing from 15 meV/atom to 10 meV/atom, depending on the amount of the database use for training.

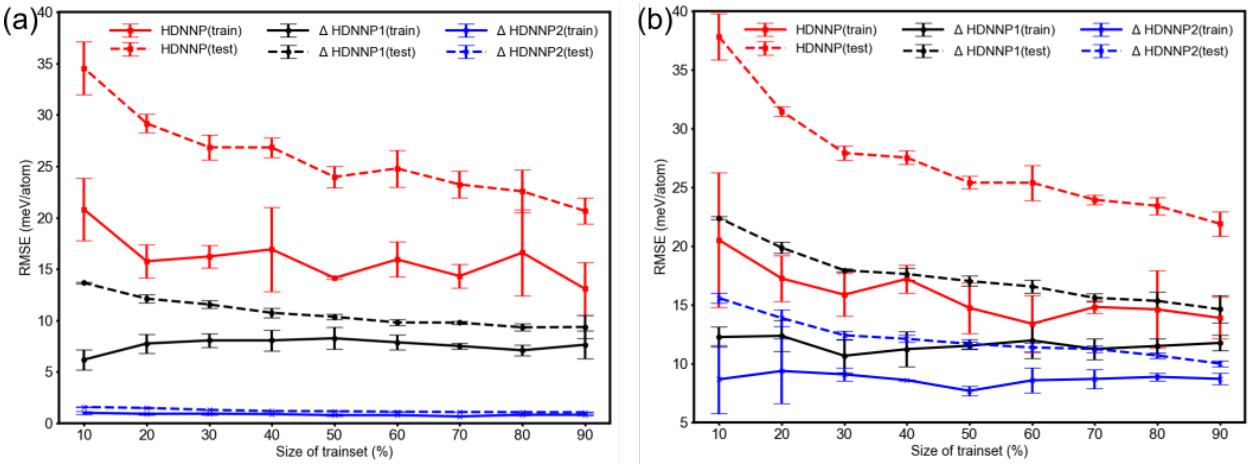

Figure 10 Using ΔHDNNP to go beyond GGA. (a) The reference method is optPBE-vdW(VASP-PW) (b) the reference method is TPSSh(VASP-PW). HDNNP (red lines) are demonstrating the results of direct train without low-level DFT. ΔHDNNP1 (black lines) shows the results of ΔHDNNP using PBE(CP2K-SZ) as low level method. ΔHDNNP2 (blue lines) shows the results of ΔHDNNP using PBE(VASP-PW) as low level DFT method. x axis is the percentage of trainset in the database and all the other structures (not used in trainings) are used as testing set. Sold lines are RMSEs of trainsets. Dash lines are RMSEs of control sets.

Figure 10 shows that the hybrid functional TPSSh is slightly more difficult to train compared with the vdW functional optPBE-vdW. The difficulty mainly comes from the exact exchange introduced in TPSSh functional, because the difference between PBE and TPSS (non-hybrid version) is much easier to train (see Figure S5). One has to use PBE(VASP-PW) calculation as the

approximate level in order to reduce the test error below 10 meV/atom. Nevertheless, the huge difference in the CPU efficiency between GGA-PBE and hybrid functional calculation still ensures the significant advantage of combing GGA-PBE plus ΔHDNNP over brute-force hybrid functional calculations in terms of the CPU time. A simple ΔHDNNP correction reduces the error in GGA-PBE from 162.2 meV/atom (Figure S3(b)) to 10 meV/atom (Figure 10) when hybrid functional is the target accuracy.

5 Hierarchical delta neural network potential

Considering the high transferability of ΔHDNNP for the energy differences shown in Figure 10, we were inspired to generalize the ΔHDNNP approach to a multi-layer case, i.e. Δ_sHDNNP (s >1). One can consider for example the more accurate but more CPU demanding non local functional optPBE-vdW(VASP-PW) as reference energy (indicated by the first bar in Figure 11(a)). Several different training schemes can be considered. i) The first method consists in a direct training of HDNNP against optPBE-vdW(VASP-PW) energies, which is the conventional way to use HDNNP. This is illustrated in the last bar in Figure 11(a). ii) The second method trains a neural network potential against the difference between optPBE-vdW(VASP-PW) and a cheap DFT method PBE(CP2K-SZV). The energy of the reference is the sum between a cheap calculation from PBE(CP2K-SZ) and a calculation from the neural network potential. This is illustrated in the third bar of Figure 11(a). Because only one layer of Δ database is used in this case, this method is called Δ_1HDNNP. iii) The third method exploits two layers of Δ database. One NN is trained against the difference between optPBE-vdW(VASP-PW) and an auxiliary database at the PBE(VASP-PW) level, which corresponds to NN2 in the fourth bar in Figure 11(a). The second NN is trained against the difference between PBE(VASP-PW) and PBE(CP2K-SZ), which corresponds to NN1 in the fourth bar of Figure 11. This approach contains two layers of Δ database, therefore it is labeled as Δ_2HDNNP. The final energy will be predicted by combining the approximate DFT (CP2K-SZ) and two Δ levels. In the productive simulation, the auxiliary level calculation is not required anymore. The method (iii) takes advantage of fact that the energy difference between optPBE-vdW(VASP-PW) and PBE(VASP-PW) can be efficiently trained with only a very small number of expensive optPBE-vdW calculations. Hence, this approach, that we call hierarchical delta neural network potential, can strongly reduce the CPU cost for the generation of the database, by decreasing the number of expensive high-level calculations.

We further benchmarked the accuracy of the described multi-layer HDNNP (Δ_sHDNNP) with the out-of-sample data collected from previous MD simulations (Figure 8) and the results are shown in Figure 11. We can see that the direct HDNNP (with trainset equal to 2126) gives an average error of 25.7 meV/atom. ΔHDNNP, which uses PBE(CP2K-SZV) as low level method and 90 % percent of the optPBE-vdW(VASP-PW) reference data (2126 structures), provides a much better accuracy and shows an average error of 8.9 meV/atom. Δ_2HDNNP gives a comparable accuracy with RMSE equal to 9.0 meV/atom. In this test, Δ_2HDNNP combines two level of ΔHDNNP. The first level describes the difference between PBE(VASP-PW) and PBE(CP2K-SZ), and uses 40 % of the PBE(VASP-PW) references (2560 structures). The second level accounts for the difference between PBE(VASP-PW) and optPBE-vdW(VASP-PW), and only uses 10 % percent of the optPBE-vdW(VASP-PW) references (236 structures). Since the optPBE-vdW(VASP-PW) functional is at least 10 times more expensive than PBE functional (shown later), Δ_2 HDNNP uses eventually less CPU time to build the trainset, but results in a very similar prediction accuracy like Δ_1HDNNP. In this example, because of the small optPBE-vdW(VASP-PW) database used in the Δ_2 HDNNP method, the two-layer Δ_2 HDNNP approach uses 23% of CPU time (according to the benchmarks in Figure 12) to build the reference but achieves much better accuracy compared with HDNNP(Figure 11(a)). Please note that the advantage of Δ_1HDNNP is that it does not require PBE(VASP-PW) database to train the first layer of Δ_1HDNNP. In practical applications, the choice between Δ_1HDNNP and Δ_2HDNNP is open and it depends on the specific system and CPU-efficiency for different xc functional.

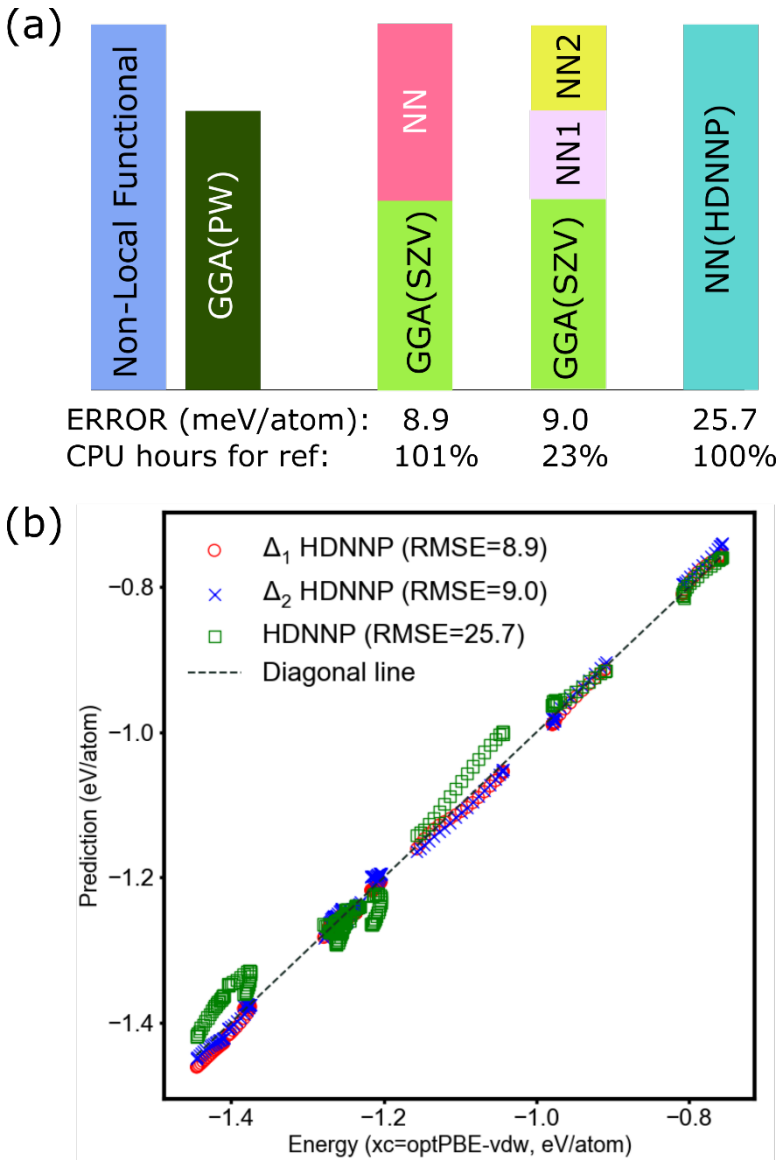

Figure 11. (a) An illustration of the hierarchical delta neural network methods. The ERROR is calculated based on the numerical experiments explained in the manuscript and the data in Figure 11(b). The CPU time required for building the reference is calculated based on the benchmarks in Figure 12. In this figure, relative values are shown. (b) Comparison of the accuracy of different training methods i.e. HDNNP, Δ_1HDNNP, and Δ_2HDNNP (see main text and Figure 11(a)) against the optPBE-vdW(VASP-PW) database. The unit for the RMSE is meV/atom. NN stands for neural network.

Compared with optPBE-vdW(VASP-PW), PBE(VASP-PW) is not an ideal intermediate level for TPSSh(VASP-PW), the trained error between PBE(VASP-PW) and TPSSh(VASP-PW) is about 10 meV/atom with 90 % database as trainset (Figure 10). Nevertheless, ΔHDNNP with either PBE(CP2K-SZ) or PBE(VASP-PW) as low accuracy DFT is still much better than the direct HDNNP, and the transferability is also significantly improved (see Figure S9). Considering the

hybrid function is much more expensive than pure GGA, the one layer ΔHDNNP is still very useful when the hybrid functional is mandatory for the accuracy.

6 CPU efficiency of DFT calculations with different basis sets and different functionals.

In order to compare the time efficiency of DFT calculation with different levels of basis sets as well as different functionals (single zeta basis set vs plane wave basis sets in this work and PBE function vs hybrid or vdW functionals), we exploited GPAW package to conduct the numerical experiments. Because GPAW packages provides both *LCAO* mode using atomic basis sets and *PW* mode using plane wave basis sets, one can get rid of the impacts from other influences like code compiling. First, the converged energy cutoff for plane waves in GPAW *PW* mode is determined as 450 eV as shown in Figure S8. Then the ratio of wall time to complete a single point calculation for different sizes of clusters are evaluated. The result is shown in Figure 12(a). For the small cluster (N=6), *PW* mode is 10 times more expensive than an approximate calculation with *LCAO* mode with single zeta basis sets. The difference becomes even more significant (30 times) for larger atoms (N=20). GPAW also provides another *FD* (finite difference) mode, in which one can reduces the grid spacing *h* to increase the accuracy of DFT calculations systematically in the same spirit of increasing basis sets. We also compared LCAO mode and FD mode to show the cost to use converged 'basis sets' compared to the approximate LCAO-SZ method. Results are shown in Figure 12(a). FD mode and PW mode in GPAW demonstrate similar CPU efficiency, which are 10~30 times more expensive than DFT with single zeta basis sets. Since all the other factors are the same in this numerical experiment, we can conclude that by using small basis set as low DFT calculation, one can gain up to 30 times speedup for a large Pt_{20} cluster and at least 10 times speedup for a small cluster (N=6).

The CPU efficiency with different exchange correlation functional is also investigated using VASP package. In this numerical experiment, VASP with plane waves up to 250 eV is exploited. Two different exchange correlation functionals are compared with pure GGA functional (PBE), one is non-local van der Waals density functional optPBE-vdW and another is hybrid functional TPSSh. The results are shown in Figure 12(b). The dependence of wall time ratio on cluster size is different for optPBE-vdw and TPSSh. The extra cost for vdW functional one results from the evaluating the non-local interactions, which is mainly dominated by the volume of the system. Therefore, the

simulation box with small cluster (N=6) has larger vacuum/cluster volume ratio is relatively more expensive compared with pure PBE. CPU cost of hybrid functional instead is mainly impacted by the number of electrons, which is obviously more expensive for larger clusters. Therefore, the computational cost of expensive functional strongly depends on the cluster size ranges from 10~30 times more than pure GGA functional.

Finally, if we compiled the effects of basis sets and xc functional, the overall improvement by using ΔHDNNP approach can be as large as 900 times faster, which provides significant advantages and opens the access to high accuracy simulations for large clusters.

Although we can achieve a significant speed-up by the ΔHDNNP approach, especially when we aim at DFT level with expensive functionals, the application of current ΔHDNNP approach for very large systems is still limited by the unfavorable scaling in CPU time with size originating from the approximate level DFT calculations. Although we did not conduct numerical experiments for very large systems, we can get some clues from the paper of Schutt, O. et al.[62] They exploit a machine learning method to construct an adaptive single zeta basis set showing an accuracy of double-zeta basis sets and the speed-up is about 50 for a bulk water with 6192 molecules. Since the acceleration mainly originates from the different sizes of the basis sets, we can also expect similar speed-up for such a system by the current ΔHDNNP approach. The 50-fold speed-up is significant, but calculations remain more expensive than empirical force field method. Further improvement on the approximate level calculation should be explored.

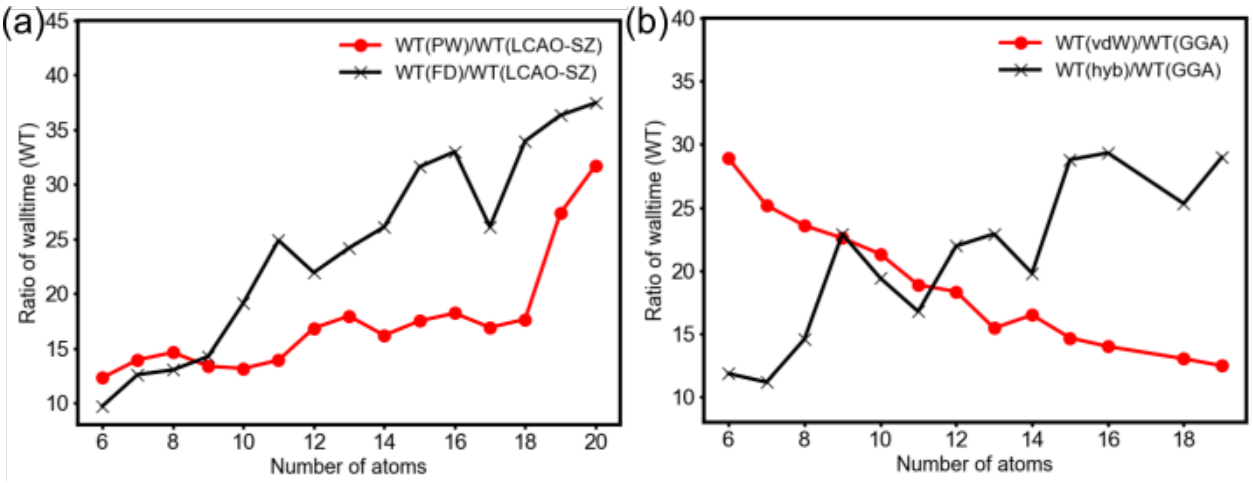

Figure 12: Comparison of the wall time (WT) for completing a single point calculation by different methods. Left (a): shows the effects of basis sets comparing converged plane waves basis sets (PW)/converged finite difference grid (FD) (see Figure S8) and single-zeta atomic basis sets. (a) is evaluated by the GPAW package with PBE functional. Right (b) shows the effects of xc functionals comparing the pure GGA methods and non-local vdw functional as well as hybrid functional. (b) is calculated by VASP package with plane wave basis sets. All the calculations are conducted on Hoffman2 at UCLA with 4 CPUs on intel-E5330 processors.

7 Conclusion

Neural network potentials are more and more popular in atomistic simulations because of their time efficiency as well as their potential high accuracy versus the reference DFT methods. However, practical applications are generally hindered by burdensome training procedures, which requires significant manual interventions and large trainset in order to achieve useful accuracy and transferability. In this contribution, we present a simple differential Δ_sHDNNP approach, using one or more layers combined with fast single zeta DFT to deliver much better accuracy: the neural network is trained to reproduce the difference between accurate energies and fast approximate ones. On the selected example of Pt clusters of size 6-20, the transferability of the Δ_sHDNNP approach is significantly improved, compared to the direct training, because the correspondence between input structural descriptors and energy is smoother. As a result, the requirement on the size of the trainset size is also minimized. Although the training is performed on clusters smaller than 20 atoms, the relative error for the energy prediction on larger clusters is very good (5.5 meV/atom for Pt_{55}) for the differential approach, while the direct training is giving large errors.

The key advantages of the Δ_sHDNNP approach include a user-friendly and easy access to accurate machine learning potential method. Especially, the required trainset is reduced by at least one order

of magnitude and Δ_sHDNNP also opens the access to simulations of large clusters with the accuracy level of expensive xc-functionals. Finally, the current approach can be generalized to other type of machine learning methods available in the community.

8 Acknowledgements

This work used computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Institute for Digital Research and Education's Research Technology Group. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. P. S. thanks UCLA for startup funding. G.S and P.S. acknowledge funding through the project "Ensemble representation for the realistic modeling of cluster catalysts at heterogeneous interfaces" by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under Award No. DE-SC0019152.

Supporting Information. Selection of descriptor parameters; energy comparison from different functionals and basis sets; force quality and energy quality from HDNNP and ΔHDNNP; convergence of basis sets in GPAW; transferability of ΔHDNNP targeting at TPSSh functional; range of symmetry function values in trainset/testset; PCA analysis of the input layer/hidden layer values.

9 Reference

1. Shaw, D. E., et al., Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330*, 341-6.
2. Boes, J. R.; Groenenboom, M. C.; Keith, J. A.; Kitchin, J. R., Neural Network and Reaxff Comparison for Au Properties. *Int. J. Quantum Chem.* **2016**, *116*, 979-987.
3. Senftle, T. P., et al., The Reaxff Reactive Force-Field: Development, Applications and Future Directions. *npj Comput. Mater.* **2016**, *2*, 15011.
4. Botu, V.; Batra, R.; Chapman, J.; Ramprasad, R., Machine Learning Force Fields: Construction, Validation, and Outlook. *J. Phys. Chem. C* **2017**, *121*, 511-522.
5. Huang, B.; von Lilienfeld, O. A., Communication: Understanding Molecular Representations in Machine Learning: The Role of Uniqueness and Target Similarity. *J. Chem. Phys.* **2016**, *145*, 161102.
6. Li, Z.; Kermode, J. R.; De Vita, A., Molecular Dynamics with on-the-Fly Machine Learning of Quantum-Mechanical Forces. *Phys. Rev. Lett.* **2015**, *114*, 096405.
7. Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A., Big Data Meets Quantum Chemistry Approximations: The Delta-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087-96.
8. Suzuki, T.; Tamura, R.; Miyazaki, T., Machine Learning for Atomic Forces in a Crystalline Solid: Transferability to Various Temperatures. *Int. J. Quantum Chem.* **2017**, *117*, 33-39.
9. Behler, J., Perspective: Machine Learning Potentials for Atomistic Simulations. *J. Chem. Phys.* **2016**, *145*, 170901.

10. Behler, J., First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems. *Angew. Chem. Int. Ed.* **2017**, *56*, 2-15.
11. Huan, T. D.; Batra, R.; Chapman, J.; Krishnan, S.; Chen, L.; Ramprasad, R., A Universal Strategy for the Creation of Machine Learning-Based Atomistic Force Fields. *npj Comput. Mater.* **2017**, *3*, 37.
12. Li, H.; Zhang, Z.; Liu, Z., Application of Artificial Neural Networks for Catalysis: A Review. *Catalysts* **2017**, *7*.
13. Artrith, N.; Hiller, B.; Behler, J., Neural Network Potentials for Metals and Oxides - First Applications to Copper Clusters at Zinc Oxide. *Phys. Status Solidi B* **2013**, *250*, 1191-1203.
14. Artrith, N.; Morawietz, T.; Behler, J., High-Dimensional Neural-Network Potentials for Multicomponent Systems: Applications to Zinc Oxide. *Phys. Rev. B* **2011**, *83*, 153101.
15. Artrith, N.; Urban, A., An Implementation of Artificial Neural-Network Potentials for Atomistic Materials Simulations: Performance for Tio_2. *Comput. Mater. Sci* **2016**, *114*, 135-150.
16. Behler, J., Atom-Centered Symmetry Functions for Constructing High-Dimensional Neural Network Potentials. *J. Chem. Phys.* **2011**, *134*, 074106.
17. Behler, J., Neural Network Potential-Energy Surfaces in Chemistry: A Tool for Large-Scale Simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930-55.
18. Behler, J., Representing Potential Energy Surfaces by High-Dimensional Neural Network Potentials. *J. Phys. Condens. Matter.* **2014**, *26*, 183001.
19. Behler, J., Constructing High-Dimensional Neural Network Potentials: A Tutorial Review. *Int. J. Quantum. Chem.* **2015**, *115*, 1032-1050.
20. Behler, J.; Parrinello, M., Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.
21. Khorshidi, A.; Peterson, A. A., Amp: A Modular Approach to Machine Learning in Atomistic Simulations. *Comput. Phys. Commun.* **2016**, *207*, 310-324.
22. Bartok, A. P.; Payne, M. C.; Kondor, R.; Csanyi, G., Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.
23. Peterson, A. A.; Christensen, R.; Khorshidi, A., Addressing Uncertainty in Atomistic Machine Learning. *Phys. Chem. Chem. Phys.* **2017**, *19*, 10978-10985.
24. Balabin, R. M.; Lomakina, E. I., Neural Network Approach to Quantum-Chemistry Data: Accurate Prediction of Density Functional Theory Energies. *J. Chem. Phys.* **2009**, *131*, 074104.
25. Balabin, R. M.; Lomakina, E. I., Support Vector Machine Regression (Ls-Svm)--an Alternative to Artificial Neural Networks (Anns) for the Analysis of Quantum Chemistry Data? *Phys. Chem. Chem. Phys.* **2011**, *13*, 11710-8.
26. Wu, J.; Xu, X., The X1 Method for Accurate and Efficient Prediction of Heats of Formation. *J. Chem. Phys.* **2007**, *127*, 214105.
27. Wu, J.; Xu, X., Improving the B3lyp Bond Energies by Using the X1 Method. *J. Chem. Phys.* **2008**, *129*, 164103.
28. Shen, L.; Wu, J.; Yang, W., Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks. *J. Chem. Theory Comput.* **2016**, *12*, 4934-4946.
29. Shen, L.; Yang, W., Molecular Dynamics Simulations with Quantum Mechanics/Molecular Mechanics and Adaptive Neural Networks. *J. Chem. Theory Comput.* **2018**, *14*, 1442-1455.
30. Artrith, N.; Behler, J., High-Dimensional Neural Network Potentials for Metal Surfaces: A Prototype Study for Copper. *Phys. Rev. B* **2012**, *85*.
31. Artrith, N.; Urban, A.; Ceder, G., Efficient and Accurate Machine-Learning Interpolation of Atomic Energies in Compositions with Many Species. *Phy. Rev. B* **2017**.
32. Imbalzano, G.; Anelli, A.; Giofre, D.; Klees, S.; Behler, J.; Ceriotti, M., Automatic Selection of Atomic Fingerprints and Reference Configurations for Machine-Learning Potentials. *J. Chem. Phys.* **2018**, *148*, 241730.

33. Artrith, N.; Urban, A.; Ceder, G., Constructing First-Principles Phase Diagrams of Amorphous Lixsi Using Machine-Learning-Assisted Sampling with an Evolutionary Algorithm. *J. Chem. Phys.* **2018**, *148*, 241711.
34. Kresse, G.; Furthmüller, J., Efficiency of Ab-Initio Total Energy Calculations for Metals and Semiconductors Using a Plane-Wave Basis Set. *Comput. Mater. Sci.* **1996**, *6*, 15-50.
35. Kresse, G.; Furthmüller, J., Efficient Iterative Schemes Forab Initiototal-Energy Calculations Using a Plane-Wave Basis Set. *Phys. Rev. B* **1996**, *54*, 11169-11186.
36. Kresse, G.; Hafner, J., Ab Initiomolecular Dynamics for Liquid Metals. *Phy. Rev. B* **1993**, *47*, 558-561.
37. Kresse, G.; Hafner, J., Ab Initiomolecular-Dynamics Simulation of the Liquid-Metal–Amorphous-Semiconductor Transition in Germanium. *Phys. Rev. B* **1994**, *49*, 14251-14269.
38. Kresse, G.; Joubert, D., From Ultrasoft Pseudopotentials to the Projector Augmented-Wave Method. *Phys. Rev. B* **1999**, *59*, 1758-1775.
39. Perdew, J. P.; Burke, K.; Ernzerhof, M., Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865-3868.
40. Staroverov, V. N.; Scuseria, G. E.; Tao, J.; Perdew, J. P., Comparative Assessment of a New Nonempirical Density Functional: Molecules and Hydrogen-Bonded Complexes. *J. Chem. Phys.* **2003**, *119*, 12129-12137.
41. Tao, J.; Perdew, J. P.; Staroverov, V. N.; Scuseria, G. E., Climbing the Density Functional Ladder: Nonempirical Meta-Generalized Gradient Approximation Designed for Molecules and Solids. *Phys. Rev. Lett.* **2003**, *91*, 146401.
42. Janthon, P.; Luo, S. A.; Kozlov, S. M.; Vines, F.; Limtrakul, J.; Truhlar, D. G.; Illas, F., Bulk Properties of Transition Metals: A Challenge for the Design of Universal Density Functionals. *J. Chem. Theory. Comput.* **2014**, *10*, 3832-9.
43. Park, J.; Yu, B. D.; Hong, S., Van Der Waals Density Functional Theory Study for Bulk Solids with Bcc, Fcc, and Diamond Structures. *Curr. Appl. Phys.* **2015**, *15*, 885-891.
44. Gautier, S.; Steinmann, S. N.; Michel, C.; Fleurat-Lessard, P.; Sautet, P., Molecular Adsorption at Pt(111). How Accurate Are Dft Functionals? *Phys. Chem. Chem. Phys.* **2015**, *17*, 28921-30.
45. Sun, G.; Sautet, P., Metastable Structures in Cluster Catalysis from First-Principles: Structural Ensemble in Reaction Conditions and Metastability Triggered Reactivity. *J. Am. Chem. Soc.* **2018**, *140*, 2812-2820.
46. Hutter, J.; Iannuzzi, M.; Schiffmann, F.; VandeVondele, J., Cp2k: Atomistic Simulations of Condensed Matter Systems. *WIREs Comput. Mol. Sci.* **2014**, *4*, 15-25.
47. VandeVondele, J.; Hutter, J., Gaussian Basis Sets for Accurate Calculations on Molecular Systems in Gas and Condensed Phases. *J. Chem. Phys.* **2007**, *127*, 114105.
48. van der Maaten, L.; Hinton, G., Visualizing Data Using T-Sne. *J. Mach. Learn. Res.* **2008**, *9*, 2579-2605.
49. van der Maaten, L.; Hinton, G., Visualizing Non-Metric Similarities in Multiple Maps. *Mach. Learn.* **2012**, *87*, 33-55.
50. Cubuk, E. D.; Malone, B. D.; Onat, B.; Waterland, A.; Kaxiras, E., Representations in Neural Network Based Empirical Potentials. *J. Chem. Phys.* **2017**, *147*, 024104.
51. Perdew, J. P., Climbing the Ladder of Density Functional Approximations. *Mrs Bulletin* **2013**, *38*, 743-750.
52. Chai, J. D.; Head-Gordon, M., Long-Range Corrected Hybrid Density Functionals with Damped Atom-Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10*, 6615-6620.
53. Duanmu, K.; Truhlar, D. G., Validation of Density Functionals for Adsorption Energies on Transition Metal Surfaces. *J. Chem. Theory Comput.* **2017**, *13*, 835-842.

54. Heyd, J.; Scuseria, G. E.; Ernzerhof, M., Hybrid Functionals Based on a Screened Coulomb Potential. *J. Chem. Phys.* **2003**, *118*, 8207-8215.
55. Janesko, B. G., Density Functional Theory Beyond the Generalized Gradient Approximation for Surface Chemistry. **2014**, *365*, 25-51.
56. Tran, F.; Stelzl, J.; Blaha, P., Rungs 1 to 4 of Dft Jacob's Ladder: Extensive Test on the Lattice Constant, Bulk Modulus, and Cohesive Energy of Solids. *J Chem Phys* **2016**, *144*, 204120.
57. Kang, R.; Lai, W.; Yao, J.; Shaik, S.; Chen, H., How Accurate Can a Local Coupled Cluster Approach Be in Computing the Activation Energies of Late-Transition-Metal-Catalyzed Reactions with Au, Pt, and Ir? *J. Chem. Theor. Comput.* **2012**, *8*, 3119-3127.
58. Li, R.; Odunlami, M.; Carbonniere, P., Low-Lying Pt-N Cluster Structures (N=6-10) from Global Optimizations Based on Dft Potential Energy Surfaces: Sensitivity of the Chemical Ordering with the Functional. *Comput. Theor. Chem.* **2017**, *1107*, 136-141.
59. Dion, M.; Rydberg, H.; Schroder, E.; Langreth, D. C.; Lundqvist, B. I., Van Der Waals Density Functional for General Geometries. *Phys. Rev. Lett.* **2004**, *92*, 246401.
60. Klimes, J.; Michaelides, A., Perspective: Advances and Challenges in Treating Van Der Waals Dispersion Forces in Density Functional Theory. *J. Chem. Phys.* **2012**, *137*, 120901.
61. Soini, T. M.; Genest, A.; Nikodem, A.; Rosch, N., Hybrid Density Functionals for Clusters of Late Transition Metals: Assessing Energetic and Structural Properties. *J. Chem. Theor. Comput.* **2014**, *10*, 4408-4416.
62. Schutt, O.; VandeVondele, J., Machine Learning Adaptive Basis Sets for Efficient Large Scale Density Functional Theory Simulation. *J Chem Theory Comput* **2018**, *14*, 4168-4175.